JAVATECH™

By Kevin M. Savetz

Getting the Jump on JavaScript

Start learning to add life to your Web pages

Static Web pages are a dying breed. Although not obsolete yet, it may be one day soon that a Web page that doesn't do anything just won't get a second glance. Once upon a time, all Web pages were static. CGI programs started to change that by allowing powerful searches of Web pages, sites that modified themselves based on the time of day or the phase of the moon, and even interactive games. Java, a powerful programming language, is bringing a whole new level of interactivity to the Web. Somewhere in the middle is JavaScript, a new language that's simple enough for beginning programmers to learn, but powerful enough to interact with Web pages and Java applets in ways previously not possible with any language.

JavaScript is a fledgling language. As I write this, JavaScript isn't even complete. The technical specifications of the language have been agreed upon, but no Web browsers completely support JavaScript. (JavaScript 3.0, which should incorporate the functions that are currently lacking, is due out in June, give or take a month; no firm date had been set at press time.) But that doesn't mean that no one is using it; bleeding-edge Web hackers have been experimenting with the crippled JavaScript for months, and many are impressed with what they see.

What is JavaScript?

Java and JavaScript are distinct languages, with different purposes and features. JavaScript was designed to provide an easy way for Web authors to create interactive Web pages. Java is a programming language which is used to create stand-alone applications, called applets. Unlike Java, which is meant for experienced programmers with an understanding of C++, JavaScript is a simpler "scripting" language (like dBASE and AppleScript) aimed at those with less programming experience.

Both are powerful languages, but they each offer distinct functionalities. According to Andy Augustine, vice president of technology for Frequency Graphics, a Web development and hosting service provider, "JavaScript is aimed at the people who don't have C++ experience, people who don't have much experience with CGI, but want to create some sort of interactive Web page."

"Contrary to popular misconceptions, JavaScript was not meant to be a scaled-down version of Java, nor was it intended to be a replacement for CGI (server-side) scripts. Instead, JavaScript functions as an outstanding way to enhance both," he said.

JavaScript can be used to manage user input as well as to show text, play sounds, display images, or communicate with a plug-in in response to "events" such as a mouse-click or exiting or entering a Web page.

JavaScript's syntax is loosely based on the Java language, but those who have used both languages say the similarity is very loose indeed. In fact, the closest resemblance the two languages have is their names. JavaScript was originally called Mocha while being developed at Netscape. It was later dubbed LiveScript, and when Netscape Communications partnered with Sun to develop Java, LiveScript was renamed JavaScript.

Kevin Savetz is author of *MBONE: Multicasting Tomorrow's Internet*, published by IDG books and waiting patiently on a bookshelf near you. You can visit Kevin's Web page at http://www.northcoast.com/savetz/ or reach him at savetz@northcoast.com.

JAVA VS. JAVASCRIPT

JavaScript programs are interpreted and run entirely on the client side. This means fewer hits and less processing time on the server than with Java (where applets are compiled on the server before being executed on the client) or CGI (which requires the server to do the work and rack up hits).

There are other differences between the two languages. In Java, applets are files distinct from HTML pages; with JavaScript, the code is integrated into the HTML using special tags. (This can cause problems with some Web browsers that do not properly handle JavaScript code.) Like Java, JavaScript is a cross-platform language that can work with any compatible browser.

Here's an example HTML file with an embedded JavaScript. Between the <BODY> tags, the Web browser displays a form with a space to enter a number. When the user enters a number in the field, the browser starts the script, located between the <SCRIPT> tags. The script itself makes sure the user really entered a number, and displays a dialog box if the user didn't keep the number within pre-defined limits.

```
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--- these comments should hide the script from old browsers
function checkNum(str, min, max) {
  if (str == "") {
    alert("Enter a number in the field, please.")
    return false
  }
  for (var i = 0; i < str.length; i++) {
    var ch = str.substring(i, i + 1)
    if (ch < "0" || ch > "9") {
      alert("Enter a number, please.")
      return false
    }
  }

  var val = parseInt(str, 10)
  if ((val < min) || (val > max)) {
    alert("Try a number from 1 to 10.")
    return false
  }
  return true
}

function thanks() {
  alert("Thank you.")
}
// end hiding from old browsers -->
</SCRIPT>
</HEAD>

<BODY>
<FORM NAME="ex5">
Please enter a number from 1 to 10:
<INPUT NAME="num"
  onChange="if (!checkNum(this.value, 1, 10))
    {this.focus();this.select();} else {thanks()}">
</FORM>
</BODY>
```

All this only takes one hit on the server – to download the HTML code and the embedded script. Doing this with a CGI script would require multiple hits on the server, once for the HTML file, and again to verify the input every time the user types into the form. JavaScript allows the client to perform more significant calculations without weighing down the server.

Other differences are more technical. JavaScript is object-based: its code uses built-in, extensible objects, but unlike Java, there are no classes or object inheritance. In JavaScript, object references are checked at run-time; in Java, object references must exist upon compilation. In JavaScript, variable data types are not declared, unlike in Java, where they must be declared before use.

Which language is better? Until a JavaScript interpreter is complete, there is no fair comparison. Although it was originally intended as a tool for fledgling programmers, even experienced programmers who know Java will find JavaScript useful. The languages can work together to complement each other.

Currently, the only Web browser with any support for JavaScript is Netscape 2.0, and it lacks much of the functionality promised for the final release. More than 30 other companies, including Microsoft, have agreed to include support for JavaScript in their Web browsers. With the current version of JavaScript, you can send information to the browser (such as a dialog box) and get the status of objects (such as mouse movements), but some of the most exciting functions, such as the abilities to read input streams from documents and to interact with Java applets, are still unimplemented.

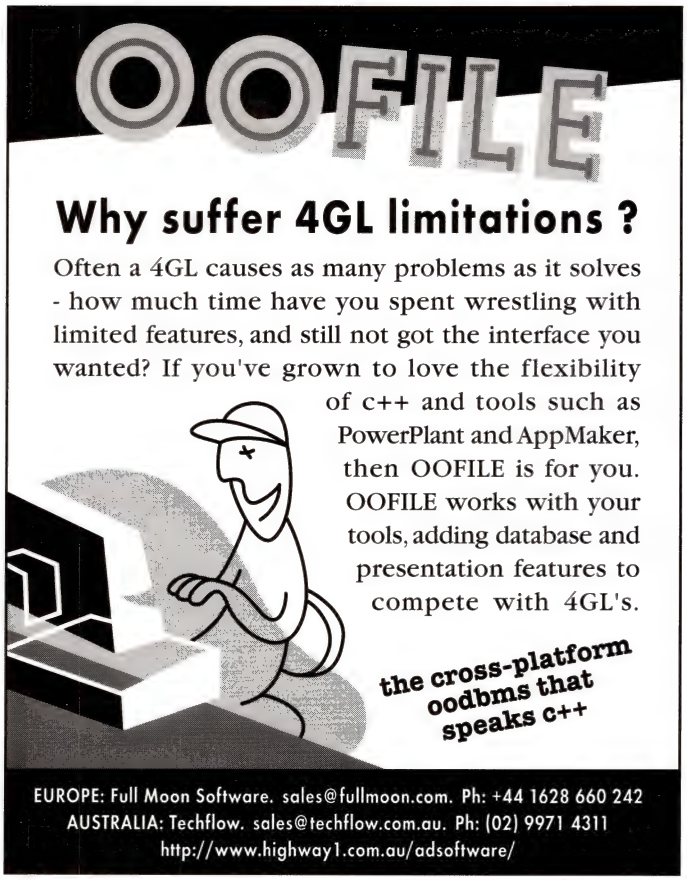
When JavaScript includes all of the tools the specs say will be included, including something called the "applet object" (which will allow scripts to interact with Java applets), the two languages will be a mighty team. "As soon as the applet object is released, you will be able to control Java applets with JavaScript, and that will make them work perfectly together," Augustine says. This means that you will be able to control a Java applet using a Web-based form, with a JavaScript doing the behind-the-scenes translation between the Web page and the applet. The user will be able to use a radio button to select how a Java applet will run – for instance, changing the color and speed of a rotating logo. JavaScript will pass information from the form to the Java applet. JavaScript's applet object will also allow scripts to send information to and get information from other scripts and CGI programs.

Another tool that will be available in the final release of JavaScript is the "history object", which will allow the script to see what Web pages the user has recently visited (although to protect the user's privacy, the script will not be able to share that information with the server). This would allow a page to offer an intelligent "back" button, letting the user move quickly back to any previously visited Web page.

As with Java, security in JavaScript is likely to be an ongoing battle. Several security problems were quickly discovered in JavaScript 2, including a bug that let JavaScript send a blank electronic mail message to someone without the sender's knowledge. Another bug allowed JavaScript to get a listing of files on the user's hard drive (although the script could not read the contents of any files).

NOW OR LATER?

Should you bother learning JavaScript now, or is it currently just a toy for experimenters? Should you wait until the final


OOFILE
Why suffer 4GL limitations ?
Often a 4GL causes as many problems as it solves - how much time have you spent wrestling with limited features, and still not got the interface you wanted? If you've grown to love the flexibility of c++ and tools such as PowerPlant and AppMaker, then OOFILE is for you. OOFILE works with your tools, adding database and presentation features to compete with 4GL's.
the cross-platform oodbms that speaks c++
EUROPE: Full Moon Software. sales@fullmoon.com. Ph: +44 1628 660 242
AUSTRALIA: Techflow. sales@techflow.com.au. Ph: (02) 9971 4311
http://www.highway1.com.au/adsoftware/

version of the JavaScript interpreter ships before trying to learn the ins and outs of the language? Augustine thinks that now is the time to get a jump start on JavaScripting. "Learn JavaScript right now if you want to capture the market while it is viable; you'll be ahead of the field. If you wait until it is fully functional before you learn it, I think you'll be behind the game," he says. Online tutorials can get you started with the basics of the language, and Internet mailing lists and newsgroups devoted to the topic are already home to thriving communities of developers and experimenters.

Will the serious Web page developer need to understand both Java and JavaScript? Because JavaScript can interact directly with the user anywhere on a Web page (rather than being limited to a single window, as with Java), most Java developers will probably want to learn both. But JavaScript should also be viable as a stand-alone tool for less experienced programmers who want to add interactivity to their Web pages.

Example Scripts

There are already dozens of example JavaScripts on the Web. Many, such as the 1040EZ tax form and body mass calculator, simply compute values input from an HTML form. Others are more inspired, such as the currency exchange calculator, which lets you select two countries from a list and then computes the exchange rate. I am particularly fond of the Zen koan generator, even though it occasionally causes Netscape 2.0 to crash.

http://www.homepages.com/fun/1040EZ.html
http://www.iohk.com/UserPages/acheng/javascript.html
http://www.superprism.net/doc/EXPRESS/util/currency.html
http://www.wrldpwr.com/javascriptzen.html

The color picker shows how JavaScript can work with frames and color: pass the mouse over the name of a color in one frame and the background of another frame will become that color. Other examples are downright inspirational, such as the JavaScript VRML clock, which uses JavaScript to display a three-dimensional clock if you have a VRML browser.

http://www.lsi.usp.br/doc/color/picker.html
http://www.phoenix.net/~jalonso/applets/vrml

Many more example applications are available at http://www.gamelan.com/.

Remember how interesting (and sometimes annoyingly funky) Web pages became when authors started using and abusing sundry new tools like backgrounds and frames? Expect to see another round of enlightening and obscure Web sites as developers and hackers begin to understand and use JavaScript.

For More Information

A growing number of information sources about JavaScript are online. Andy Augustine's JavaScript FAQ is at:

http://www.freqgrafx.com/411/jsfaq.html

There is also a JavaScript mailing list, a high-volume, sometimes technical list for discussion about using the language. To subscribe, send a message to majordomo@obscure.org with a message body of "subscribe javascript". A digest version of the list is also available. To subscribe, use a message body of "subscribe javascript-digest". Further discussion takes place on the newsgroup comp.lang.javascript.

Netscape's official JavaScript documentation is required reading if you plan to learn your way around JavaScript. The files are also downloadable for reading offline.

http://home.netscape.com/eng/mozilla/2.0/handbook/javascript/index.html
http://home.netscape.com/eng/mozilla/2.0/handbook/javascript/jsdoc.zip

Augustine maintains a list of online tutorials, example applications and other information at JavaScript 411. Finally, the ubiquitous Yahoo naturally has an index of JavaScript-related goodies.

http://www.freqgrafx.com/411/
http://www.yahoo.com/Computers_and_Internet/Languages/JavaScript/